09/855,090.

WEST Search History

DATE: Friday, August 29, 2003

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=NO; OP=ADJ		
L17	L16 and l1	16	L17
L16	selecting same (dropping or dragging) same quer$5	37	L16
L15	L13 and (l6 or l2)	44	L15
L14	L13 and (l6 or l5 or l4 or l3 or l2)	767	L14
L13	L12 and l1	767	L13
L12	(visual or search) near3 query	1871	L12
L11	l3 same l4	42	L11
L10	l3 same l5	0	L10
L9	(l1 or l3) and (l4 or l5) and (l2 or l6)	0	L9
L8	l1 and l3 and (l4 or l5) and (l2 or l6)	0	L8
L7	l1 and l2 and l3 and (l4 or l5) and l6	0	L7
L6	domain adj1 object$1	282	L6
L5	dropping near8 visual	88	L5
L4	dropping near8 query	42	L4
L3	query	21574	L3
L2	search adj1 object$1	529	L2
L1	(707/3 OR 707/104.1 OR 707/102 OR 345/700 OR 345/968).CCLS.	4923	L1

END OF SEARCH HISTORY

WEST

Generate Collection | Print

L15: Entry 3 of 44 File: USPT Jan 7, 2003

DOCUMENT-IDENTIFIER: US 6505191 B1
TITLE: Distributed computer database system and method employing hypertext linkage analysis

Abstract Text (1):
A distributed computer database system includes one or more front end computers, one or more home nodes, one or more index nodes and one or more object nodes interconnected by a network into a search engine for retrieval of hypertext documents. A query from a user is transmitted to one of the front end computers, which forwards the query to one of the home nodes, of the search engine. The home node parses the query into one or more elementary queries and schedules the elementary queries for processing. Each elementary query can be one of a number of types, including an index query, a link query or an object query. To process an index query or link query, the home node extracts features from the index query or link query, fragments the extracted features into feature fragments, and hashes these features. Each hashed feature fragment is transmitted to one index node on the network. Each index node on the network that receives a hashed feature fragment uses the hashed feature fragment of the index query or link query to perform a search on its respective partition of the database. The results of the searches of the local databases are gathered by the home node. To process an object query, the home node transmits the object identifier contained in the object query to the object node on the network containing the information associated with the object. The object node that receives the object query uses the object identifier to perform a search on its respective partition of the database. The results of the search of the local database are transmitted to the home node. The home node processes the results for each elementary query according to the specifications in the query. When all processing is completed by the home node, the results are returned to the front end node, which formats the results for presentation to the user.

Brief Summary Text (23):
More specifically, a distributed computer database system implementing the invention includes one or more front end computers, one or more home nodes, one or more index nodes and one or more object nodes interconnected by a network into a search engine for retrieval of hypertext documents. A query from a user is transmitted to one of the front end computers, which forwards the query to one of the home nodes, of the search engine. The home node parses the query into one or more elementary queries and schedules the elementary queries for processing. Each elementary query can be one of a number of types, including an index query, a link query or an object query. To process an index query or link query, the home node extracts features from the index query or link query, fragments the extracted features into feature fragments, and hashes these features. Each hashed feature fragment is transmitted to one index node on the network. Each index node on the network that receives a hashed feature fragment uses the hashed feature fragment of the index query or link query to perform a search on its respective partition of the database. The results of the searches of the local databases are gathered by the home node. To process an object query, the home node transmits the object identifier contained in the object query to the object node on the network containing the information associated with the object. The object node that receives the object query uses the object identifier to perform a search on its respective partition of the database. The results of the search of the local database are transmitted to the home node. The home node processes the results for each elementary query according to the specifications in the query. The processing may include evaluation of additional elementary queries. When all processing is completed by the home node, the results are returned to the front end node which formats the

results for presentation to the user.

Brief Summary Text (24):
In another embodiment of the invention, a distributed computer database system includes one or more front end computers, one or more home nodes, one or more index nodes and one or more object nodes interconnected by a network. A single computer processor can fulfill the functionality of one or more front end, home, index and object nodes. The combination of computer nodes interconnected by a network operates as a search engine. A user wishing to query the database, transmits the query to one of the front end nodes which in turn forwards the query to one of the home nodes of the network. The node receiving the query, termed the home node of this query, parses the query into elementary queries. For an index or link query, the home node extracts the features of the received query, fragments the features into feature fragments and then encodes the feature fragments using a hash function. A portion of each hashed feature fragment is used by the home node as an addressing index by which the home node transmits the hashed index or link query feature to an index node on the network specified by that fragment portion. For an object query, the home node uses a portion of the OID as an addressing index by which the home node transmits the object query to an object node on the network. Each index node on the network that receives a hashed index or link query feature fragment uses the hashed index or link query feature fragment to perform a search on its respective database. Index nodes finding data corresponding to a hashed index query feature fragment return the set of OIDs of the information objects possessing this feature. Index nodes finding data corresponding to a hashed link query feature fragment return the set of pairs of OIDs of the links between information objects which possess this feature fragment. Each object node on the network that receives an object query uses the OID contained in the object query to perform a search on its respective database. The object node returns the information associated with the OID as specified in the object query. Such information may include any or all of the following: the location of the object whose OID is contained in the object query, the set of OIDs that represent objects referenced by the object whose OID is contained in the object query, the set of OIDs that represent objects that reference the object whose OID is contained in the object query, and other auxiliary information associated with the object whose OID is contained in the object query. The OIDs or pairs of OIDs are then gathered by the home node. For an index or link query, a similarity function is computed based on the features that are in common with the index or link query. The similarity function is used to rank the objects or links between objects. The objects or links between objects that have the largest similarity value are used in subsequent processing of the query. For an object query, the information returned by the object node is used in subsequent processing of the query. The subsequent processing of the query by the home node may involve the construction of new elementary queries using the information returned from earlier elementary queries. Processing continues until no additional elementary queries are needed. The home node then performs any remaining processing required by the query, and the results are transmitted to the front-end node. The front end node formats the response to the user based on the OIDs and any other information transmitted by the home node. For example, if the front end node is a World Wide Web server, then the front end node constructs a page in HTML format containing a reference to a URL and auxiliary information for each object. The front end transmits the formatted response to the user.

Detailed Description Text (6):
After verifying that the query is acceptable, the front end computer 105 performs any reformatting necessary to make the query compatible with the requirements of the search engine. The front end computer 105 then transmits the query to one of the home nodes 107 of the search engine (Step 202), which is then defined as the home node of the search engine for that query.

Detailed Description Text (8):
The purpose of an object query is to obtain information about the object identified by the OID of the object query. The purpose of an index query is to search for information objects that contain information similar to the information in the index query. The purpose of a link query is to search for pairs of information objects such that one object contains information similar to the information in the link query and such that this information is associated with a link to the other information object.

Detailed Description Text (11):
Features are extracted from unstructured documents by using feature extraction algorithms, e.g., implemented as computer programs executable by the home node 215. Feature extraction produces a data structure consisting of a collection of inter-related component data structures. The data structure is divided into (possibly overlapping) substructures, as in the case of a structured document, and these substructures are the fragments of the unstructured document. A large variety of feature extraction algorithms have been developed for media such as sound, images and video streams, for example, edge detection, segmentation and object classification algorithms for images. Fourier and Wavelet transformations as well as many filtering algorithms are also used to extract features from images and sound. Video streams comprise a sequence of images and a synchronized sound track. In addition to feature extraction from the individual images and sound track, video streams can be organized into scenes; domain objects in successive images can be identified with one another and domain objects of the sound track can be related to domain objects in the corresponding scenes. Features extracted from video streams can also include these domain objects. Each feature can have one or more values associated with components of the data structure that represents the feature. In the simplest case the data structure consists of a single component with an associated value. In this case, the feature represents one attribute of the object. More complex features will contain several inter-related components, each of which may have attribute values. The data structures that represent the features conform to a data model specified by the ontology. The data model determines the kinds of components and attribute values that are allowed. Each fragment of each feature has an associated weight, representing the strength of the feature or degree to which the information object possesses the feature.

Detailed Description Text (41):
The Feature Extractor 508 extracts features from an object or elementary query. Feature extraction for images is performed by detecting edges, identifying the image objects, classifying the image objects as domain objects and determining relationships between domain objects. In another embodiment, feature extraction for images is performed by computing Fourier or wavelet transforms from images and sound. Each Fourier or wavelet transform constitutes one extracted feature. Video streams comprise a sequence of images and a synchronized sound track. In addition to feature extraction from the individual images and sound track, video streams can be organized into scenes; domain objects in successive images can be identified with one another and domain objects of the sound track can be related to domain objects in the corresponding scenes. Features extracted from video streams can also include these domain objects. The extracted features are transferred to the Fragmenter 510. In addition, when features have been extracted from an object, the features are transferred to the Communication Module 504 in the form of an Insert Object Message.

Current US Original Classification (1):
707/3

L15: Entry 6 of 44 File: USPT Sep 17, 2002

DOCUMENT-IDENTIFIER: US 6453312 B1
TITLE: System and method for developing a selectably-expandable concept-based search

Brief Summary Text (3):
This invention relates generally to an improved system and method for aiding users in the development of a query string; and more specifically, to a system and method for allowing a user to control the development of a concept-based natural language search query by controlling the manner in which a hierarchical concept tree is structured and traversed.

Brief Summary Text (8):
Many challenges exist when developing an information retrieval system that is capable of aiding users in finding meaningful information from a large body of electronic data. Often times, the users of such systems are only familiar with general topics of interest, and are not able to specify the actual terminology used within the textual information that is relevant for a desired topic. Additionally, the user may not be familiar with the keyword descriptors used to index the documents. As a result, the user-provided query may be incomplete or inaccurate. Other factors, for example regional language dialects, may further influence the construction of a search query. For all of these reasons, the information retrieved during a search may only yield a small number of documents that are actually relevant.

Brief Summary Text (9):
One way to improve search results is to provide a mechanism for automatically expanding a user-provided query string to include terms that do not appear in the query, but which may correspond to, or be associated with, the user-provided query terms. A system of this type is disclosed in U.S. Pat. No. 5,265,065 to Turtle. This patent describes a system in which words of a natural language input query are replaced with phrases from a database in a manner that expands the query. The expanded query is then utilized for information retrieval. The problem with many prior art systems of this type is that no opportunity is provided for the user to interactively participate in the query development process. If the search is unsuccessful, the user must simply re-execute the search with a different query string.

Brief Summary Text (10):
One method which does provide an iterative technique for allowing a user to interactively refine a query is provided in U.S. Pat. No. 5,278,980 to Pedersen et al. This patent describes a process whereby a user-provided query is developed into a search string that is further used to locate a list of matched phrases from a corpus of documents. Words from the returned phrases that are not included in the original query can be used to refine the query. This process can be repeated to retrieve documents that are increasingly focused on the desired topic. Although this system provides an opportunity for the user to exert control over the query development, the user intervention is only allowed after a search has been performed. This may waste processing resources, and does not provide any insight into the manner in which the user-provided query is developed into the actual search string.

Brief Summary Text (15):
What is needed is a flexible search system that allows users to closely control the expansion of a search query. The system should also provide the capability for a user to modify the manner in which particular query expansions. This capability should include the ability to modify both the system lexicon, and the lexicon organization. The system should allow users to add user-specific terminology such as regional

jargon, slang terms, foreign language representations, or any other particularized phrases included in the body of material to be searched.

Brief Summary Text (18):
It is another object of the invention to provide a system that allows users to interactively control the manner in which a search query is expanded;

Brief Summary Text (19):
It is yet another object of the invention to provide a system wherein an iterative, interactive process is utilized to allow a user to expand a search query while exploring the lexicon of the system;

Brief Summary Text (25):
It is yet another object of the invention to utilize an object-oriented repository to implement a hierarchical concept tree for use in interactive query development, and wherein a query developed by the system may be utilized to search other objects stored in the object-oriented repository.

Brief Summary Text (30):
The system includes a user interface that allows for interactive traversal of the various relationships in the hierarchical tree structure. The word and word variant elements located during this traversal are added to a potential query string. Traversal of the tree structure begins by locating a user-provided character string in one of the word or word variant elements. Traversal of the tree continues so that related concepts are located, and further, so that all other word and word variants related to the located concepts are also located. The user is allowed to select or de-select each of the located concepts, words, or word variants for further inclusion in the query development process. Traversal of the hierarchical tree structure continues with the parents and children of the remaining initially-located concepts. After each additional level in the concept tree is traversed, the user is allowed to again specify selection or de-selection of any of the located concepts, words, or word variants. The user is also allowed to specify whether the search should be expanded to include parents, children, or siblings of a previously-located concept. This allows the user to expand query development to include concepts that would otherwise not be located during traversal of the parents and children of the initially-located concept. This iterative process continues until a selected number of levels in both the parent and child directions have been traversed in the hierarchical tree structure for the initially-located concepts. This query development method allows the user to view search terms as the terms are added to the query, and further allows control over search expansion in a manner which is not provided for in prior art systems. Because the user controls the concepts that will be included in the search query, a single search may include multiple related, yet distinct concepts.

Brief Summary Text (31):
After traversal of the hierarchical tree structure has progressed to the extent specified by the user, the word and word variant strings related to the selected concepts may be formed into a query string that includes logical operators. The query string will be formatted as required by the search tools that will receive the query string. The query string is provided to manually or programmatically invoke any number of various tools used to perform a text or file search. These tools include text editors, web-based search engines, file management systems, or object management systems used to catalog and track the development of software constructs.

Brief Summary Text (32):
According to one embodiment of the invention, the hierarchical tree structure is implemented in an object-oriented repository. The developed query string is used to search other objects stored within the same object-oriented repository.

Drawing Description Text (12):
FIG. 10 is a block diagram showing how a query string developed by Search String Wizard may be utilized;

Detailed Description Text (18):
Element Inventory Scheme (EIS) 103 is a model that defines the various element types and relationship types that can be created and stored within Object Management System.

The elements and relationships that are created according to this model can then be used to create a hierarchical tree structure storing concepts and word indicators that can be used to develop a search query. Before discussing search query development, the EIS model is described in more detail.

Detailed Description Text (22):
The Element type "Locator Element" 208 and all element types which are a subtype of element type "Locator Element" 208 are the element types used to develop a search query, and are of primary interest to the current invention. Within the defined hierarchy, element type "Locator Element" has several sub-types, including element types "Application Domain" 218, "Concept" 220, "Word" 222, and "Word Variant 224". By virtue of the subtype relationship represented by Line 225, each of these subtypes inherit the characteristics of element type "Locator Element" 208. This includes the ability to form relationships of the type that can be formed by element type "Locator Element".

Detailed Description Text (36):
Using Locator Elements to Formulate a Search Query

Detailed Description Text (39):
FIG. 4 is an exemplary screen provided by Search String Wizard (SSW) 121 to users to obtain the search string and other optional parameters required to initiate development of a search query. The user provides the initial search string in Box 400. In this example, assume the user provides the term "Bugs". The user further selects the levels of hierarchy within the hierarchical concept tree in one or more Application Domains that will be utilized to develop the search. This is accomplished by specifying whether parent, sibling, or child concepts will be included for any located concepts. These parameters are specified in Boxes 402, 404, and 406, respectively. If parent and/or child concepts are to be included, the number of levels of hierarchy to be traversed during the search development may be specified in Boxes 408 and 410. A default value of "one" is assumed if no value is provided. Finally, the extent of automation may be specified in Box 412. If the search development is to be completely automated, Box 412 is left blank. Otherwise, "Interactive Mode" is selected, which causes SSW 121 to query the user for input following each iteration of search development. The manner in which the above-mentioned options are utilized will be discussed in the examples below.

Detailed Description Text (45):
Assume in this example the user is only concerned about developing a search query related to the concept "VW Beetle". The user employs the interactive user display to mark the concepts "Software Problem" and "Insects" for deletion, then selects "Delete Button" 506 to remove these concepts from Window 502. In a similar manner, the user is allowed to mark Word or Word Variant character strings in Window 502 for deletion. In the preferred embodiment, the user display will show the deleted Word and Word Variant character strings in Window 510, and will show the deleted Concepts in Window 511. The user is able to re-add words or concepts that have been deleted by selecting them from Windows 510 and 511, respectively, then using the "Add Button" 508 to re-add the item to the appropriate Window 502 or 504.

Detailed Description Text (51):
At this point, the additional functions shown in FIG. 5 as "Expand to Parent Concept" 516, "Expand to Sibling Concept" 518, and "Expand to Child Concept" 520 will be discussed. These functions are used after first selecting a given concept from Window 504, then selecting one or more of these functions and hitting the "Continue" Button 514. This function will find all Parent, Sibling, or Child concepts for the selected concept, and will further provide these newly-located concepts in Window 504. These functions will further locate all Word and Word Variant Elements associated with the newly-located concepts and will add these located words to the list in Window 502. These functions provide a way for a user to control the expansion of the search query in a way that does not necessarily fit into the initial search request, and in this respect might be described as "search expansion" functions. For example, assume the user wanted to determine all child Concept Elements for the Concept Element "Car". These child Concept Elements would not be located by the parameters that were initially specified as shown in FIG. 4. However, using these additional functions, the user is able to explore this portion of the concept tree to potentially find

additional relevant search terms. These functions may be utilized at any stage of query development when Interactive Mode is utilized, or may be used following search completion when the Interactive Mode is not selected.

Detailed Description Text (54):
The above description exemplifies the Interactive Mode shown selected in Box 412 of FIG. 4. If this mode is not selected using the Interactive Mode button 412, the user is not provided with an interactive display following each iteration of the search in the manner that is described above. Instead, the search query is expanded automatically in what might be described as "Non-Interactive Mode" according to the initially-selected parameters. When the search has been completely expanded, the user is provided with a screen including all located Concept, Word, and Word Variant Elements. The user is allowed to delete elements from this display, and is further allowed to utilize the "Expand to Parent Concept", "Expand to Child Concept" and "Expand to Sibling Concept" functions on any selected Concept Element as discussed above. Finally, the user may select the Continue button 514 so that the terms may be formulated into a search string including logical operators in the manner described above.

Detailed Description Text (56):
According to yet another embodiment of the invention, the user may invoke SSW 121 in a batch mode with the input parameters shown in FIG. 4. Using batch mode, a complete set of search terms is located such as that shown in FIG. 8. In this mode, the finished query may be written to a file, or may be programmatically returned to the calling search tool. The query may thereafter be processed by a script running on Script Server 142 to programmatically insert logical operators into the query. For example, script commands can be utilized to programmatically insert a logical operator "OR" between each of the search terms included in the query.

Detailed Description Text (63):
Once developed by the SSW 121, the query string may be used in a number of different ways. It may be provided via a communications connection to a web-based search engine running on any local or remote server. If the search engine has the capability to be programmatically invoked, the search string may be supplied to the search engine without human intervention. Alternatively, a user may copy the search string from a file or "clipboard" and manually provide it to the search engine. The search string may then be used in searching web pages, for example. Any search engine may use the previously-developed query string, as long as the string is formatted with the appropriate syntax as required by the search engine. The SSW will be loaded with one or more rule sets so that the query string may be developed with the appropriate syntax based on the selected target search tool that is to receive the query string.

Detailed Description Text (68):
FIG. 10 is a block diagram providing various examples of utilizing a query string developed by Search String Wizard (SSW) 121. Search String Wizard 121 is an add-in component that can be installed for use with any number of Search-Specific Environments that are shown as Box 1002. First SSW will be loaded with one or more sets of rules to inform SSW of the desired syntax to be used in developing the search. The syntax is based on the particular Search-Specific Environment. This initialization step is shown by Arrow 1004. Then SSW may be invoked from the Search-Specific Environment, as shown by Arrow 1006.

Detailed Description Text (69):
In the embodiment described above, Element Locator 120 is used to invoke SSW 121. However, the tool could also be invoked using interfaces from other search tools. For example, a Web-based search browser represented by Block 1008 could be utilized. Alternatively, a text editor represented by Block 1010, or a file manager illustrated as Block 1012, could be used to invoke the add-in component. Once invoked, the user will specify the input parameters and the desired mode of operation. SSW will retrieve Locator Elements from Element Repository 101 via service calls to AIM EXE 105, as shown by Arrow 1014. The requested elements are returned to SSW, as illustrated by Arrow 1016, so the search query can be formatted according to user specifications, including the use of any logical operators. Thereafter, the search string may be programmatically returned to the Search-Specific Environment in Block 1002 as depicted by Arrow 1018, or the search string may be manually copied, for example, by use of a

"clip-board". In one embodiment, the search string will be provided to Element Locator 120, which uses the search string to make service calls to AIM EXE 105 to search the Attribute Fields of Asset Elements. This is shown by Arrow 1020. The located Asset Elements are thereafter returned to Element Locator 120 as shown by Arrow 1022. Alternatively, the search string may be used by the web-based search browser shown in Block 1008 , the text editor as shown in Block 1010, or a file manager as shown in Block 1012, to search a group of files, web pages, or text documents as represented by Block 1024. This use of the query string is shown by Arrows 1026 and 1028.

Current US Original Classification (1):
707/3

CLAIMS:

22. The method of claim 16, and further including the step of providing said query string to a search tool for use in performing a search.

L15: Entry 10 of 44 File: USPT Feb 12, 2002

DOCUMENT-IDENTIFIER: US 6347313 B1
TITLE: Information embedding based on user relevance feedback for object retrieval

Abstract Text (1):
A method and system for indexing and retrieving database objects, such as images, include a database manager which initializes database objects based on vectors for values of quantified features associated with the database objects. Similar database objects are grouped into common clusters that are based on system-perceived relationships among the objects. For each search session, a vector for a search query is calculated and database objects from the closest cluster within feature space are selected for presentation at a user device. The user indicates which of the selected objects are relevant to the search session and which of the objects are irrelevant. If one of the clusters includes both relevant and irrelevant objects, the cluster is split into two clusters, so that one of the resulting clusters includes the relevant objects and the other cluster includes irrelevant objects. The correlation matrix is updated to indicate that the resulting clusters have a weak correlation. If two of the clusters include database objects which were indicated to be relevant to the search session, the correlation matrix is updated to indicate that the two clusters have a strong correlation. To avoid an excessive proliferation of database clusters, mergers are performed on clusters which are closely located within the feature space and share a strong correlation within the correlation matrix. Following continued use, the groupings of objects into clusters and the cluster-to-cluster correlations will reflect user-perceived relationships.

Detailed Description Text (31):
Although the method has been described above as being practiced on database images, any type of database file can be substituted for image files including, but not limited to, audio files, text files, spreadsheet files or graphics files. Additionally, while the method has been described as using query objects to initiate a search, this is not critical. As one alternative, a randomized selection of images from various clusters may be presented in a first iteration of a search session, with the user-designations of relevance and irrelevance being employed to narrow search results in the subsequent iterations of the session. As another alternative, the search may be initiated by a word description that is converted to a feature vector.

Current US Original Classification (1):
707/3

CLAIMS:

1. A method of managing objects of a database comprising steps of:

presenting first database objects in response to a search query of a first search session;

enabling a user to designate a first subset of said first database objects as being relevant to said first search session and to designate a second subset of said first database objects as being irrelevant to said first search session;

at least partially based upon designations of said first and second subsets, organizing said objects of said database into multi-object groups and defining correlations among said groups, organization of a plurality of said objects into a specific said group being indicative of content similarities among said objects, said

correlations being indicative of user-perceived similarities among said groups; and

storing said organization of said groups and said defined correlations among said groups for access during a second search session following completion of said first search session.

8. The method of claim 7 wherein said reconfiguring step further includes formulating membership functions for a query object included in said search query, each query object membership function describing a degree of membership of said query object within one of said groups.

16. The system of claim 13 wherein said configuration module is configured to assign a high correlation of similarity to an association between a first object group and a third object group which both include database objects that have been determined to be relevant to a first search query object.

19. A method of managing objects within a database utilizing user feedback comprising steps of:

configuring said objects into a network of clusters based on values of quantified features of said objects and based on correlations among said clusters;

detecting a first search query of a search session;

selecting a first set of objects determined to be responsive to said first search query, including basing said selecting upon said values and said correlations, said first set including objects from each of first and second clusters;

embedding user feedback into a correlation matrix that defines said correlations among said clusters in said network, including:

(a) assigning a high correlation to linking said first and said second clusters if objects within each of said first and second clusters are designated by a user to be relevant to said search session; and

(b) splitting said first cluster into a third and fourth cluster if said first cluster is designated by said user to include both relevant and irrelevant objects with respect to said search session; and

storing said correlation matrix after embedding said user feedback for use in subsequent search sessions.

23. The method of claim 19 further comprising the step of decreasing a correlation assigned to said first and said second clusters if objects from one of said first and said second cluster are designated as being relevant to said search and objects from the other of said first and said second cluster are designated as being irrelevant to said search.

WEST

Generate Collection | Print

L15: Entry 16 of 44 File: USPT Apr 24, 2001

DOCUMENT-IDENTIFIER: US 6223145 B1
TITLE: Interactive interface for specifying searches

Abstract Text (1):
An interactive interface for creating a search query for a corpus of machine-readable documents, each of which is associated with at least one category of a category hierarchy. The interactive interface includes a cone tree generation component, a query specification component, a begin search component, and a query generation component. The cone tree generation component generates and displays a cone tree representing the category hierarchy. The cone tree represents each category of the category hierarchy as a node having a selection object for indicating inclusion of the category in a one of a first group, and a second group. Each selection object is responsive to a cursor control device. The query specification component generates and displays a query specification object including a first group object and a second group object. Each group object is responsive to the cursor control device and indicates members of the group. Each group includes a one of a term included within the corpus and a category of the category cone tree. The begin search component generates and displays a begin search object responsive to the cursor control device. The query generation component generates a query for searching the corpus to find documents that include at least one member of the first group and at least member of the second group.

Brief Summary Text (5):
Increasingly, the public accesses text collections and their information electronically, identifying desired documents by submitting a search query. The standard approach for displaying search results for electronic text collections is to present, in ranked order, a list of document titles. Often shown alongside each title is a numerical score signifying the degree of match between the document and the query, or the estimated relevance of the document to the query. On-line bibliographic systems show meta-data about the documents, such as author and publisher, alongside the title. Search engines associated with the World Wide Web commonly show short summaries or excerpts from the retrieved documents, typically extracted from the first few lines. Some systems show lines extracted from the document that match terms in the query. Document titles can also be annotated with graphics that show the correspondence between the retrieved documents and the query.

Brief Summary Text (43):
The interactive interface of the present invention allows users to create a search query for a corpus of machine-readable documents, each of which is associated with at least one category of a category hierarchy. The interactive interface includes a cone tree generation component, a query specification component, a begin search component, and a query generation component. The cone tree generation component generates and displays a cone tree representing the category hierarchy. The cone tree represents each category of the category hierarchy as a node having a selection object for indicating inclusion of the category in a one of a first group, and a second group. Each selection object is responsive to a cursor control device. The query specification component generates and displays a query specification object including a first group object and a second group object. Each group object is responsive to the cursor control device and indicates members of the group. Each group includes a one of a term included within the corpus and a category of the category cone tree. The begin search component generates and displays a begin search object responsive to the cursor control device. The query generation component generates a query for searching the corpus to find documents that include at least one member of the first group and at

least member of the second group.

Drawing Description Text (16):
FIG. 14 illustrates instructions for using the interactive interface of the present invention to specify a search query.

Detailed Description Text (2):
FIG. 10 illustrates in block diagram form computer system 30 in which the interactive user interface of the present invention is implemented by executing instructions 200. Instructions 200 alter the operation of computer system 30 enabling presentation of an interactive user interface for specifying search queries for a document corpus associated with a large category hierarchy. Briefly described, the interactive interface uses information visualization technology to better and separately represent a categorical hierarchy. The interactive interface of the present invention includes a Cat-a-Cone, a query specification object, and a begin search object. The Cat-a-Cone, as suggested by its name, is a cone tree that separately represents a categorical hierarchy independent of a corpus of documents with which it is associated. The Cat-a-Cone represents each category of the category hierarchy as a node, each of which has a selection objection to indicate whether the category is to be included as a member of a first group or a second group. The query specification object looks like a book cover and includes a multiplicity of group objects, or group windows. Using a mouse or other cursor control device, the user may insert into the group object the text terms to be grouped together. While a group object is active, the user may also use the mouse to select categories of the Cat-a-Cone to be included in the same group. The user indicates the query is complete by clicking on the begin search object, or search button. In response, the system generates a search query for all documents in the corpus that include at least one member of each group. Afterward, the search query is presented to a search engine.

Detailed Description Text (13):
Execution of instructions 200 begins with step 202, during which processor 31 generates a query construction object and displays it on monitor 32. The query construction object uses a book cover metaphor, which cover includes several windows, to facilitate query construction. Subsequently, during step 204, processor 31 also generates and displays on monitor 32 a search button. Preferably, the search button is displayed such that it appears to be part of the query construction object or appears very near the query construction object. By activating the search button the user signals that query construction is complete and that a search of the corpus for documents satisfying the query should begin. Finally, during step 206 processor 31 displays the Cat-a-Cone associated with the selected corpus of documents, which was previously generated during step 192. Preferably, the Cat-a-Cone is displayed near the query construction object to facilitate its use in query construction.

Detailed Description Text (19):
Preferably, each group object 220 and its selection unit is associated with a unique color, to allow users to quickly visually distinguish between the different groups of the search query. In this embodiment, when a category 215 has been selected for inclusion in a group 220 its query selection unit 213 is then displayed in the color associated with that group. Alternately, each group object 220 and its associated selection unit 222 may be associated with a unique visual symbol. In this embodiment, when a category 215 has been selected for inclusion in a group 220 then the display of the associated query selection unit 213 includes the visual symbol unique to that group 220.

Detailed Description Text (20):
The present interactive interface also includes search button 218. Search button 218 responds to activation or a "click on it" by taking the text terms and categories associated with each group object 220, constructing a query that is a conjoining of disjuncts, and submitting that query to the associated search engine. Search button 218 may be displayed on query construction object 216 or near it.

Detailed Description Text (22):
FIG. 14 illustrates in flow diagram form steps 240 for using the interactive interface 210 to construct a search query. First, as shown in step 242, the user determines the number of groups 220 she wishes to be conjoined together. She then selects the

disjuncts, or members, of each group 220. As discussed above, the user does so by activating a group object 220, typing in the desired text terms via keyboard 34 and by "clicking on" the desired categories 215.

Detailed Description Text (30):
During step 302 processor 31 first examines the documents retrieved and identifies the categories associated with each document. Processor 31 uses this information to "prune" the Cat-a-Cone 214 used during the search, thus creating retrieval Cat-a-Cone 214a. That is to say, processor 31 eliminates from the Cat-a-Cone 214 representing the entire corpus the representation of those categories that are neither ancestors, nor descendants, of the categories associated with the retrieved documents. The number of categories associated with the retrieved documents is quite likely to be greater than the number of categories in the search query when multiple category labels are associated with the documents of a corpus. In other words, if each document j has c.sub.j categories assigned to it, and the search returns d.sub.i documents, then a total of approximately d.sub.i *(cj-1) categories will be associated with the d.sub.i documents retrieved. Step 302 thus enables the interactive interface to show only the relevant categories of the Cat-a-Cone 214a while maintaining the context of those categories within the hierarchy.

Detailed Description Text (43):
While processor 31 animates the ruffling of the pages of SearchBook 225, during step 412 processor 31 also animates the rotation and modification Cat-a-Cone 214a. As the pages of SearchBook 225 are moved from one side of the book to the other during the ruffle, processor 31 modifies the categories displayed by Cat-a-Cone 214 so that the categories associated with the currently displayed link page 226 are displayed. This process might be thought of as "ruffling" Cat-a-Cone 214a. Ruffling Cat-a-Cone 214a shows the user how tightly or loosely distributed the category labels are for each document retrieved by the search query and how widely distributed the categories are between the retrieved documents.

Current US Cross Reference Classification (1):
707/3

CLAIMS:

1. An interactive interface for creating a search query on a display controlled by a processor coupled to a cursor control device and to a memory storing instructions for implementing the interactive interface, the processor searching a corpus of documents in machine-readable form, each document being associated with at least one category of a category hierarchy, the interactive interface comprising:

a) a cone tree generation component for generating and displaying on the display a cone tree representing the category hierarchy, each category of the category hierarchy being represented as a node having a selection object for indicating inclusion of the category in a one of a first group and a second group, each selection object being responsive to the cursor control device;

b) a query specification component generating and displaying on the display a query specification object including a first group object and a second group object, each group object being responsive to the cursor control device and for indicating members of the group, each group including a one of a term to be searched for within the corpus and a category of the category cone tree;

c) a begin search component for generating and displaying on the display a begin search object responsive to the cursor control device, the begin search component responding to the cursor control device by indicating that the query is complete; and

d) a query generation component coupled to the begin search object, the query specification component and the corpus, the query generation component generating a query for all documents in the corpus and that include at least one member of the first group and at least one member of the second group.

2. In a computer system including a display, a cursor control device, and a processor executing instructions stored in memory, a computer implemented method of specifying a

search query over a corpus of machine-readable documents, each document of the corpus being associated with at least one category of a category hierarchy, the method comprising:

a) generating and displaying on the display a category cone tree visually representing the category hierarchy, each category of the category hierarchy being represented as a node having a selection object for indicating inclusion of the category in a one of a first group and a second group, each selection object being responsive to the cursor control device;

b) generating and displaying on the display a query specification object including a first group object and a second group object, each group object being responsive to the cursor control device and for indicating members of the group, each group including a one of a term included within the corpus and a category of the category cone tree;

c) generating and displaying on the display a begin search object responsive to the cursor control device to indicate the query specification is complete;

d) responding to indication via the cursor control device that the query specification is complete by generating a query for all documents in the corpus that include at least one member of the first group and at least one member of the second group.

L17: Entry 8 of 16 File: USPT Mar 27, 2001

DOCUMENT-IDENTIFIER: US 6208985 B1
**** See image for Certificate of Correction ****
TITLE: Data refinery: a direct manipulation user interface for data querying with integrated qualitative and quantitative graphical representations of query construction and query result presentation

Detailed Description Text (34):
To define the refine operation then, the user first selects one of the fields from the Fields list 406, and then selects one of the values for that field in the Values list 408. Once selected in this manner, the refine operation may be instantiated as either a filter operation or a tag operation, depending on where it is placed in the query construction area 200. The user moves the selected criterion into the query construction area 200 using one of two methods: (1) using a drag-and-drop implementation by dragging the selected value from the Values list 408 with the mouse or other pointing device to an icon bay 208, and releasing the mouse at that location, or (2) an alternative method, wherein the user right-clicks the mouse to employ the shortcut menu 410, and selecting one of the listed icon bays 208. The drag-and-drop methodology is made available through Microsoft Corp.'s Windows95 application programming interface. In either cases, the Data Refinery 108 automatically generates the appropriate icon 206 corresponding the target icon bay 208, and instantiates the corresponding refine (filter or tag) operation with the selected field and value. Each refine operation has at least one criterion, including a field, a operator or qualifier, and a value. In the resulting refine operation, the field is the entity or attribute selected from the Fields list 406, the operator is by default "=," and the value is the value selected in the Values list 408. The refine operation uses a default equals operator, since it is assumed that the user intends to refine based on items that have a field value equal to the selected field and value.

Current US Original Classification (1):
707/3

WEST

Generate Collection | Print

L17: Entry 10 of 16 File: USPT Jan 11, 2000

DOCUMENT-IDENTIFIER: US 6014661 A
TITLE: System and method for automatic analysis of data bases and for user-controlled dynamic querying

Detailed Description Text (45):
At any time after the system has determined the type and ranges of the various data fields, the system proceeds with query device selection. Consider once again FIG. 2. In the preferred embodiment of the invention, the initially presented query device will depend primarily on how many different possible values a data field can assume. The thresholds for selecting the different query devices will be predetermined and pre-programmed into the system, but can be changed under user control after initial presentation (for example, by activating a icon of the desired query device and then "dragging" it to the currently displayed query device, by activating and selecting from a pull-down menu adjacent to the currently displayed device, or by using any other known technique for changing portions of a graphical user interface). For example, if the data type if Boolean, a toggle may be predetermined to be the initial query device selected. For string and/or numerical data with fewer than, for example, seven different values, a checkbox or pull-down menu may be the default query device. For fields (variables) with more than some predetermined threshold number of different values, however, the default query device may be a single or range slider, depending on the data type.

Current US Original Classification (1):
707/3